

CLAIMS

What is claimed is:

1 1. A process for providing a representation of specified characteristics of a
2 previously developed object-oriented software program, said program including a number of
3 object classes and further including object related methods belonging to respective classes,
4 said process comprising the steps of:

5 sensing at least one complex method call included in said software program, a
6 plurality of said methods being associated with each of said complex method calls;

7 extracting a number of single method calls from each of said complex method calls;

8 generating a set of information for each of said methods from said single method
9 calls, the information set for a particular method containing at least the name of the particular
10 method and the class to which the particular method belongs; and

11 constructing a representation of interactions between objects of said software program
12 from the information contained in said method information sets.

1 2 The process of Claim 1 wherein:

2 said extracting step comprises replacing a component of a complex method call with
3 a phase variable to produce a method call of reduced complexity.

1 3. The process of Claim 2 wherein:

2 said process includes an initial step of extracting the name and class of each of said
3 methods from said software program.

1 4. The process of Claim 1 wherein:

2 a given complex method call comprises multiple method related components, and
3 said extracting step comprises recursively substituting a phase variable for each of said
4 method related components, until said given complex method call has been resolved into
5 multiple lines, each containing one of said single method calls.

1 5. The process of Claim 4, wherein said extracting step comprises:

2 a first parsing phase disposed to separate any casting operations included in said
3 given complex method call;

4 a second parsing phase disposed to isolate any method parameters included in said
5 given complex method call; and

6 a third parsing phase disposed to resolve any continuous method calls included in said
7 given complex method call into multiple lines, each containing one of said single method
8 calls.

1 6. The process of Claim 5 wherein:

2 said first parsing phase is implemented prior to said second parsing phase, and said
3 second parsing phase is implemented prior to said third parsing phase.

1 7. The process of Claim 6 wherein:

2 said step of generating method information sets includes parsing an output provided
3 by said third parsing phase to determine the correct object class for each of said object related
4 methods.

1 8. The process of Claim 7 wherein:

2 said process includes the step of determining whether a method is a user-defined or a
3 standard API method.

1 9. The process of Claim 1 wherein:

2 said constructing step comprises constructing a sequence diagram depicting the
3 interactions between respective objects of said software program.

1 10. The process of Claim 1 wherein:

2 the sequence diagram displays the condition of a method call to indicate that the call
3 occurs only when the condition is evaluated to be true.

1 11. The process of Claim 1 wherein:

2 said software program is in the form of source code.

1 12. The process of Claim 1 wherein:

2 said software program is written in Java software code.

1 13. The process of Claim 1 wherein:

2 said software program is written in C++ software code.

1 14. The process of Claim 1 wherein:

2 at least one of said object related methods in said program is a polymorphic method..

1 15. The process of Claim 1 wherein:

2 at least one of said object related methods in said program is related to an inheritance
3 feature, and said extraction step includes tracking an inheritance path until it reaches a parent
4 class wherein the method is defined.

1 16. A system for providing a representation of specified characteristics of a
2 previously developed object-oriented software program, said program including a number of
3 object classes and object related single methods belonging to respective classes, said program
4 further including at least one complex method call containing a plurality of said single
5 methods, said system comprising:

6 a Method Detail Parser unit disposed to extract a number of individual method calls
7 from each of said complex method calls;

8 a data base operable to store a set of information for each of said single methods, the
9 information set for a particular single method containing at least the name of the particular
10 method and the class to which the particular method belongs; and

11 a drawing device operable to construct a representation of interactions between
12 objects of said software program from the information contained in said method information
13 sets.

1 17. The system of Claim 16 wherein:

2 said system includes a Method Information Parser unit disposed to extract the name
3 and class of each of said single methods from said software program.

1 18. The system of Claim 17 wherein:

2 said Method Detail Parser unit is disposed to recursively substitute a placeholder variable
3 for each of a plurality of method related components contained in a given complex method
4 call, until said given complex method call has been resolved into multiple lines, each
5 containing one of said single method calls.

1 19. The system of Claim 18, wherein:

2 said Method Detail Parser unit is sequentially operated to implement a first parsing
3 phase to separate any casting operations included in said given complex method call, to
4 implement a second parsing phase to isolate any method parameters included in said given
5 complex method call, and to implement a third parsing phase to resolve said given complex
6 method call into multiple lines, each containing one of said single method calls.

1 20. The system of Claim 19 wherein:

2 said drawing device is operable to construct a sequence diagram depicting the
3 interactions between respective objects of said software program.

1 21. The system of Claim 20 wherein:

2 said software program is in the form of source code.

1 22. Apparatus for providing a sequence diagram representing specified
2 characteristics of a previously developed object-oriented software program, said program
3 including a number of object classes and further including object related methods belonging
4 to respective classes, said apparatus comprising:

5 means for sensing at least one complex method call included in said software
6 program, a plurality of said methods being associated with each of said complex method
7 calls;

8 Method Detail Parser means for extracting a number of single method calls from each
9 of said complex method calls;

10 means for generating a set of information for each of said object related methods from
11 said single method calls, the information set for a particular object related method containing
12 at least the name of the particular method and the class to which the particular method
13 belongs; and

14 means for constructing sequence diagram representing interactions between objects of
15 said software program from the information contained in said method information sets.

1 23. The apparatus if Claim 22 wherein:

2 said apparatus includes Method Information Parser means for extracting the name and
3 class of each of said methods from said software program.

1 24. The apparatus of Claim 23 wherein:

2 said Method Detail Parser means is operable to recursively substitute a phase variable
3 for each of a plurality of method related components contained in a given complex method
4 call, until said given complex method call has been resolved into multiple lines, each
5 containing one of said single method calls.

1 25. The apparatus of Claim 24, wherein:

2 said Method Detail Parser means is disposed to separate any casting operations
3 included in said given complex method call during a first parsing phase, to isolate any
4 method parameters included therein during a second parsing phase, and resolve said given
5 complex method call into multiple lines, each containing one of said single method calls,
6 during a third parsing phase.

1 26. The apparatus of Claim 25 wherein:

2 said first parsing phase is implemented prior to said second parsing phase, and said
3 second parsing phase is implemented prior to said third parsing phase.

1 27. The apparatus of Claim 26 wherein:

2 said constructing means comprises a drawing engine for depicting interactions
3 between respective objects of said software program.

1 28. The apparatus of Claim 27 wherein:

2 said software program is in the form of source code.